Journal of Applied Logic 9 (2011) 23–40

Contents lists available at ScienceDirect

Journal of Applied Logic

www.elsevier.com/locate/jal

Reasoning about conditional probabilities in a higher-order-logic theorem prover

Osman Hasan *, Sofiène Tahar

Department of Electrical and Computer Engineering, Concordia University, 1455 de Maisonneuve W., Montreal, Quebec, H3G 1M8, Canada

ARTICLE INFO

Article history:
Received 17 July 2010
Accepted 13 December 2010
Available online 8 January 2011

Keywords:
Binary channel
Bayes theorem
Formal methods
HOL theorem prover
Probabilistic analysis
Total probability law

ABSTRACT

In the field of probabilistic analysis, the concept of conditional probability plays a major role for estimating probabilities when some partial information concerning the result of the experiment is available. This paper presents a higher-order-logic definition of conditional probability and the formal verification of some classical properties of conditional probability, such as, the total probability law and Bayes' theorem. This infrastructure, implemented in the HOL theorem prover, allows us to precisely reason about conditional probabilities for probabilistic systems within the sound core of HOL and thus proves to be quite useful for the analysis of systems used in safety-critical domains, such as space, medicine and transportation. To demonstrate the usefulness of our approach, we provide the precise probabilistic analysis of the binary asymmetric channel, a widely used concept in communication theory, within the HOL theorem prover.

1. Introduction

Probabilistic analysis is a tool of fundamental importance for the analysis of hardware and software systems. These systems usually exhibit some random or unpredictable elements. Examples include failures due to environmental conditions or aging phenomena in hardware components and the execution of certain actions based on a probabilistic choice in randomized algorithms. Moreover, these systems act upon and within complex environments that themselves have certain elements of unpredictability, such as noise effects in hardware components and the unpredictable traffic pattern in the case of telecommunication protocols. Due to these random components, establishing the correctness of a system under all circumstances usually becomes impractically expensive. The engineering approach to analyze a system with these kind of unavoidable elements of randomness and uncertainty is to use probabilistic analysis. The main idea behind probabilistic analysis is to mathematically model the random and unpredictable elements of the given system and its environment by appropriate random variables. The probabilistic properties of these random variables are then used to estimate the probabilities associated with events of interest, such as downtime, availability, number of failures, capacity, and cost. Thus, instead of guaranteeing that the system meets some given specification under all circumstances, the probability that the system meets this specification is reported.

Today, simulation is the most commonly used computer based probabilistic analysis technique. Most simulation software provide a programming environment for defining functions that approximate random variables for probability distributions. The random elements in a given system are modeled by these functions and the system is analyzed using computer simulation techniques [15], such as the Monte Carlo method [34], where the main idea is to approximately answer a query on a probability distribution by analyzing a large number of samples. Due to the inherent nature of simulation, the proba-

* Corresponding author.
E-mail addresses: o_hasan@ece.concordia.ca (O. Hasan), tahar@ece.concordia.ca (S. Tahar).

1570-8683/$ – see front matter
doi:10.1016/j.jal.2011.01.001

bilistic analysis results attained by this technique can never be termed as 100% accurate. Moreover, simulation requires an enormous amount of CPU time for attaining meaningful estimates. We generally need to acquire hundreds of thousands of samples to estimate the desired probabilistic quantities and this fact makes the simulation approach impractical when each sample acquisition step involves extensive computations.

The precision and accuracy of hardware and software system analysis results has become imperative these days because of the extensive usage of these systems in safety and financial critical areas, such as medicine, transportation and stock exchange markets. One of the unfortunate incidents, related to the inaccurate probabilistic analysis of systems, is the loss of the Mars Polar Lander [36] in December 1999. The Mars Polar Lander; a $165 million NASA spacecraft launched to survey Martian conditions, is believed to be lost mainly because of its engine shutdown while it was still 40 meters above the Mars surface. The engine shutdown happened due to the vibrations caused by the deployment of the lander's legs, i.e., a random behavior, that gave false indication that the spacecraft had landed. Some other such incidents related to inaccurate or inadequate probabilistic analysis of systems include the loss of $125 million Mars Climate Orbiter [35] in 1998 and the faulty operation of the fly-by-wire primary flight control software of a Boeing 777, operated by the Malaysia Airlines, in August 2005 [9], which could have resulted in the loss of 177 passenger lives if the pilot had not manually taken over the autopilot program in time. In order to avoid incidents like the ones mentioned above, simulation should not be solely relied upon for the analysis of hardware and software systems that are supposed to be used in safety-critical domains.

The inaccuracy problem of simulation can be resolved by modeling the system behavior, including its random components, in a precise logic and reasoning about probabilistic and statistical properties of the system using this precise model in a mechanical theorem prover [10]. In this approach, random components are represented in the logic as random variables, which are basically functions that map the sample space of a random process to the real numbers. Higher-order logic [17] is expressive enough to be able to model such functions and thus can be used in the above mentioned probabilistic analysis approach. In fact, most of the commonly used random variables have been formalized in higher-order logic and their corresponding probabilistic properties have been verified using interactive theorem proving techniques, see [29,22,23] for example. This available formalization has been successfully used to conduct precise probabilistic analysis of a number of real-world probabilistic analysis problems like the famous Stop-and-Wait protocol or the Coupon Collector's problem [30,26–28,25]. These results clearly demonstrate the usefulness and scalability of higher-order-logic theorem proving in the probabilistic analysis domain. But, the fact that the higher-order-logic formalization of many core probabilistic analysis related mathematical foundations, such as stochastic process theory or Lebesgue integration theory, is still not available somewhat downplays its effectiveness. Thus, despite its benefits in terms of the precision of results, higher-order-logic theorem proving is rarely used for probabilistic analysis as system designers and engineers do not want to indulge in the formalization and verification tasks associated with core mathematical concepts.

In order to strengthen the higher-order-logic library of probabilistic analysis related mathematical foundations, this paper presents an approach for formal reasoning about *conditional probabilities* of events in a theorem prover. Conditional probability is one of the most important concepts in probability theory. It refers to the probability of an event A in the sample space S, given the occurrence of some other event B, also in S, and is mathematically defined as follows [32]

$$Pr(A|B) = \frac{Pr(A \cap B)}{Pr(B)} \tag{1}$$

provided $0 < Pr(B)$, where Pr represents the probability function. Conditional probabilities are quite frequently used in system analysis for estimating probabilities when some partial information concerning the result of the experiment is available. The basic definition of Markov chains, which is one of the most widely used modeling technique in the area of probabilistic analysis, is also based on the concept of conditional probabilities. Thus, the ability to formally reason about conditional probabilities in a theorem prover would definitely allow us to handle a broader range of systems and properties in the higher-order-logic theorem proving based probabilistic analysis approach, which to the best of our knowledge lacks this capability up till now.

In this paper, we first present a higher-order-logic formalization of the definition of conditional probability given in Eq. (1). Based on this formal definition, we also present the verification of some classical properties of conditional probability in the HOL theorem prover [18]. The primary motive behind selecting the HOL theorem prover is to be able to build upon the existing probabilistic analysis related higher-order-logic formalization. The formally verified properties of conditional probability not only ensure the correctness of our definition of conditional probability but also play a vital role in conducting probabilistic analysis of systems involving conditional probabilities. These properties can be reused to reason about conditional probabilities of a system and thus speed up the analysis process. In order to demonstrate the practical effectiveness of the infrastructure for reasoning about conditional probabilities, developed in this paper, we utilize it to conduct the probabilistic analysis of a binary asymmetric channel [47], which is a widely used communication channel model in coding and information theories.

The rest of the paper is organized is as follows: Section 2 provides a review of related work. Then, in Section 3, we present some preliminaries including a brief introduction to the HOL theorem prover and an overview of modeling random variables in higher-order logic and verifying their probabilistic properties in HOL. Next, in Section 4, we present our higher-order-logic definition of conditional probability and the formal verification of some of its classical properties using the HOL theorem prover. This is followed by the probabilistic analysis of the binary asymmetric channel in Section 5. Finally, we

draw some conclusions in Section 6 and highlight the areas which can be targeted as potential future work based on the infrastructure presented in this paper.

2. Related work

Due to inaccuracies introduced by the simulation based probabilistic analysis methods, many researchers around the world are exploring the usage of formal methods [19] for probabilistic analysis. Probabilistic model checking [5,44] is a rapidly emerging formal probabilistic analysis technique. Like traditional model checking, probabilistic model checking involves the construction of a precise state-based mathematical model of the given random system, which is then subjected to exhaustive analysis to verify if it satisfies a set of probabilistic properties formally expressed in some appropriate logic. Numerous probabilistic model checking algorithms and methodologies have been proposed in the open literature, e.g., [1, 39], and based on these algorithms, a number of tools have been developed, e.g., PRISM [33] and VESTA [45]. Besides the accuracy of the results, other promising features of probabilistic model checking include the ability to perform the analysis automatically and to formally verify conditional probabilities related to systems. On the other hand, probabilistic model checking is limited to systems that can be expressed as probabilistic finite state machines or Markov chains. Another major limitation of the probabilistic model checking approach is state space explosion [12]. Similarly, to the best of our knowledge, it has not been possible to precisely reason about statistical quantities, such as expectation and variance, using probabilistic model checking so far. Some probabilistic model checkers, such as PRISM [33] and VESTA [45], offer the capability of verifying expected values in a semi-formal manner. For example, in the PRISM model checker, the basic idea is to augment probabilistic models with cost or rewards: real values associated with certain states or transitions of the model. This way, the expected value properties, related to these rewards, can be analyzed. But it is important to note that the meaning ascribed to the these properties is, of course, dependent on the definitions of the rewards themselves and thus there is always some risk of verifying false properties. Similarly, due to the state-based nature of model checking techniques, none of the model checkers can verify generic mathematical expressions for statistical properties like expectation and variances, because of their continuous nature.

The higher-order-logic theorem proving based probabilistic analysis approach, utilized in this paper, tends to overcome the limitations of both the simulation and model checking based probabilistic analysis approaches. Due to the formal nature of the models and properties and the inherent soundness of the theorem proving approach, probabilistic analysis carried out in this way is free from any approximation and precision issues. Similarly, the high expressiveness of higher-order logic allows us to analyze a wider range of systems without any modeling limitations, such as the state-space explosion problem or the limitation to Markovian chain models.

The early foundations of probabilistic analysis in a higher-order-logic theorem prover were laid down by Nedzusiak [38] and Bialas [8] when they proposed a formalization of some measure and probability theories in higher-order logic. Hurd [29] implemented their work and developed a framework for the verification of probabilistic algorithms in the HOL theorem prover. Random variables are basically probabilistic algorithms and thus can be formalized and verified, based on their probability distribution properties, using the methodology proposed in [29]. In fact, building upon Hurd's formalization, most of the commonly used discrete [29] and continuous [22] random variables have been formalized in higher-order-logic and their corresponding probabilistic [24] and statistical [23] properties have been verified using interactive theorem proving techniques. These results have been successfully used to conduct precise probabilistic analysis of a number of applications, such as computation algorithms [30,26], real-time systems [27], communication protocols [25] and wireless systems [28]. In this paper, we extend the above mentioned formalization infrastructure available in the HOL theorem prover with the ability to formally reason about conditional probabilities, a novelty that to the best of our knowledge has not been presented in the open literature so far.

An alternative method for probabilistic verification in higher-order logic has been presented by Audebaud et al. [4]. Instead of using the measure theoretic concepts of probability space, as is the case in Hurd's approach, Audebaud et al. based their methodology on the monadic interpretation of randomized programs as probability distributions. The monads used in this approach are restrictive than the ones used in Hurd's approach but suffice for the given purpose. Audebaud's approach uses functional and algebraic properties of the unit interval and has been successfully used to verify a sampling algorithm of the Bernoulli distribution and the termination of various probabilistic programs in the Coq theorem prover [13]. The infrastructure in this approach is not mature enough to be able to formally analyze real-world probabilistic analysis problems as the mathematical foundations to support reasoning about continuous random variables and statistical properties does not exist so far. Therefore, in this paper we have chosen to build upon Hurd's formalization, which has more foundational probabilistic analysis theories available.

3. Preliminaries

In this section, we provide a brief introduction to the HOL theorem prover and present an overview of Hurd's methodology [29] for the verification of probabilistic algorithms. The intent is to introduce fundamental concepts along with some notations that are going to be used in the rest of the paper.

Table 1
HOL symbols and functions.

HOL symbol	Standard symbol	Meaning
∧	*and*	Logical *and*
∨	*or*	Logical *or*
¬	*not*	Logical *negation*
⇒	*implies that*	Logical *implication*
(a, b)	$a \in A \wedge B \in B \Rightarrow (a, b) \in A \times B$	A pair of two elements
fst	fst (a, b) = a	First component of a pair
snd	snd (a, b) = b	Second component of a pair
λx.t	$\lambda x.t$	Function that maps x to $t(x)$
{x|P(x)}	$\{x \mid P(x)\}$	Set of all x such that $P(x)$
{x|𝕋} *or* UNIV	$\mathbb{U}$	Universal set
{x|𝔽} *or* {}	$\emptyset$	Empty set
compl A	$\acute{A}$	Complement of set A
A subset B	$A \subseteq B$	A is a subset of B
A inter B	$A \cap B$	A intersection B
A union B	$A \cup B$	A union B
A diff B	$A - B$	Difference between sets A and B
disjoint A B	$A \cap B = \emptyset$	Sets A and B are disjoint
image f A	$\{f(x) \mid x \in A\}$	Set with elements $f(x)$ for all $x \in A$
bigunion P	$\{x \mid \exists S.\ S \in P,\ x \in S\}$	Union of all sets in the set P
sum(0, k)(λn.f(n))	$\sum_{n=0}^{k-1} f(n)$	Sum of first k terms of sequence f
suminf(λn.f(n))	$\lim_{k\to\infty} \sum_{n=0}^{k} f(n)$	Infinite summation of f
summable(λn.f(n))	$\exists x. \lim_{k\to\infty} \sum_{n=0}^{k} f(n) = x$	Summation of f is convergent

3.1. HOL theorem prover

The HOL theorem prover is an interactive theorem prover which is capable of conducting proofs in higher-order logic. It utilizes the simple type theory of Church [11] along with Hindley–Milner polymorphism [37] to implement higher-order logic. HOL has been successfully used as a verification framework for both software and hardware as well as a platform for the formalization of pure mathematics.

In order to ensure secure theorem proving, the logic in the HOL system is represented in the strongly-typed functional programming language ML [41]. An ML abstract data type is used to represent higher-order-logic theorems and the only way to interact with the theorem prover is by executing ML procedures that operate on values of these data types. The HOL core consists of only 5 basic axioms and 8 primitive inference rules, which are implemented as ML functions. Soundness is assured as every new theorem must be verified by applying these basic axioms and primitive inference rules or any other previously verified theorems/inference rules.

HOL supports two types of interactive proof methods: forward and backward. In forward proof, the user starts with previously proved theorems and applies inference rules to reach the desired theorem. In most cases, the forward proof method is not the easiest solution as it requires the exact details of a proof in advance. A backward or a goal directed proof method is the reverse of the forward proof method. It is based on the concept of a *tactic*, which is an ML function that breaks goals into simple subgoals. In the backward proof method, the user starts with the desired theorem or the main goal and specifies tactics to reduce it to simpler intermediate subgoals. Some of these intermediate subgoals can be discharged by matching axioms or assumptions or by applying existing decision procedures. The above steps are repeated for the remaining intermediate goals until we are left with no further subgoals and this concludes the proof for the desired theorem.

The HOL theorem prover includes many proof assistants and automatic proof procedures [20] to assist the user in directing the proof. The user interacts with a proof editor and provides it with the necessary tactics to prove goals while some of the proof steps are solved automatically by the automatic proof procedures.

In order to facilitate reutilization of verified theorems, HOL allows its users to store a collection of valid HOL types, constants, axioms and theorems as a HOL theory file. Once stored, HOL theories can be loaded in the HOL system and the corresponding definitions and theorems can be utilized right away. Thus, HOL theories allow us to build upon existing results in an efficient way without going through the tedious process of regenerating these results using the basic axioms and primitive inference rules. Various mathematical concepts have been formalized and saved as HOL theories by the HOL users. Out of this useful library of HOL theories, we utilized the theories of Booleans, lists, sets, positive integers, *real* numbers, measure and probability in this paper. In fact, one of the primary motivations of selecting the HOL theorem prover for our work was to benefit from these built-in mathematical theories.

Table 1 provides the mathematical interpretations of some frequently used HOL symbols and functions, which are inherited from existing HOL theories, in this paper.

Table 2
Formally verified probability axioms.

Axiom	HOL theorem
Probability Empty	$\texttt{prob}\ \{\texttt{s} \mid \mathbb{F}\} = 0$
Probability Universe	$\texttt{prob}\ \{\texttt{s} \mid \mathbb{T}\} = 1$
Probability Bounds	$\forall \texttt{A}.\ 0 \leqslant \texttt{prob(A)} \leqslant 1$
Probability Complement	$\forall \texttt{A}.\ \texttt{prob(compl A)} = 1 - \texttt{prob(A)}$
Probability Increasing	$\forall \texttt{A B}.\ \texttt{A subset B} \Rightarrow \texttt{prob(A)} \leqslant \texttt{prob(B)}$
Probability Additive	$\forall \texttt{A B}.\ (\texttt{disjoint A B}) \Rightarrow \texttt{prob(A union B)} = \texttt{prob(A)} + \texttt{prob(B)}$
Probability Countably Additive	$\forall \texttt{A i}.(\forall \texttt{i j}.(\texttt{i} \neq \texttt{j}) \Rightarrow (\texttt{disjoint A}_{\texttt{i}}\ \texttt{A}_{\texttt{j}})) \Rightarrow \texttt{prob(bigunion A}_{\texttt{i}}\texttt{)} = \sum_{\texttt{i}} \texttt{prob(A}_{\texttt{i}}\texttt{)}$

3.2. Probabilistic analysis in HOL

Hurd [29] formalized some measure theory in higher-order logic to define a measure space as a pair (Σ, μ). In this formalization, the sample space is the universal set of the appropriate type. Building upon this formalization, the probability space was also defined as a pair $(\mathcal{E}, \mathbb{P})$, where the domain of $\mathbb{P}$ is the set $\mathcal{E}$, which is a set of subsets of infinite Boolean sequences $\mathbb{B}^\infty$. Both $\mathbb{P}$ and $\mathcal{E}$ are defined using the Carathéodory's Extension theorem, which ensures that $\mathcal{E}$ is a σ-algebra: closed under complements and countable unions.

Now, a random variable, which is one of the core concepts in probabilistic analysis, is fundamentally a probabilistic function and thus can be modeled in higher-order logic as a deterministic function, which accepts the infinite Boolean sequence as an argument. These deterministic functions make random choices based on the result of popping the top most bit in the infinite Boolean sequence and may pop as many random bits as they need for their computation. When the functions terminate, they return the result along with the remaining portion of the infinite Boolean sequence to be used by other programs. Thus, a random variable which takes a parameter of type α and ranges over values of type β can be represented in HOL by the function.

$$\mathcal{F} : \alpha \to B^\infty \to \beta \times B^\infty$$

As an example, consider the *Bernoulli*$(\frac{1}{2})$ random variable that returns 1 or 0 with equal probability $\frac{1}{2}$. It can be formalized in HOL as follows

```
⊢ bit = λs.  (if shd s then 1 else 0, stl s)
```

where *s* is the infinite Boolean sequence and `shd` and `stl` are the sequence equivalents of the list operations '*head*' and '*tail*'. Now, we first focus upon the optimization of the above formalization of the *Bernoulli*$(\frac{1}{2})$ random variable and will then present the verification of our formalization, i.e., to formally prove that the function `bit` indeed models the *Bernoulli*$(\frac{1}{2})$ random variable.

The probabilistic programs can be expressed in the more general state-transforming monad [6] where the states are the infinite Boolean sequences.

```
⊢ ∀a s. unit a s = (a,s)
⊢ ∀f g s. bind f g s = g (fst (f s)) (snd (f s))
```

The `unit` operator is used to lift values to the monad, and `bind` is the monadic analogue of function application. All monad laws hold for this definition, and the notation allows us to write functions without explicitly mentioning the sequence that is passed around, e.g., function *bit* can be defined as

```
⊢ bit_monad = bind sdest  (λb. if b then unit 1 else unit 0)
```

where `sdest` gives the head and tail of a sequence as a pair `(`*shd* `s`, *stl* `s)`.

The second step in conducting probabilistic analysis of a system using a higher-order-logic theorem prover is to use the formal model of the given system to express the probabilistic properties regarding the events of interest in the system as higher-order-logic theorems and formally verify these theorems. For the verification task we usually need some basic probability axioms and probabilistic properties of the random variables that are used to construct the systems model. The formalized `prob` and $\mathcal{E}$ can be used to derive all the basic axioms of probability in the HOL theorem prover. For example, some of the formally verified probability axioms are given in Table 2.

The formalized `prob` and $\mathcal{E}$ can also be used to prove probabilistic properties for random variables. For example, we can formally verify the following probabilistic property for the function `bit`, defined above,

$$\vdash \texttt{prob}\ \{\texttt{s} \mid \texttt{fst (bit s)} = 1\} = \frac{1}{2}$$

where the function `fst` selects the first component of a pair. The above theorem verifies that the function `bit` essentially models a *Bernoulli*$(\frac{1}{2})$ random variable.

As mentioned in Section 2, the infrastructure described above has been successfully used to model most of the discrete and continuous random variables and verify their probabilistic and statistical properties. In this paper, we mainly extend this infrastructure with the ability to reason about conditional probabilities.

4. Conditional probabilities in HOL

The notion of conditional probability involves the probability of an event, say A, given the information that an event B has occurred. In order to understand the usage of conditional probability, consider the example of an integrated circuit testing process where we test circuits in pairs. The concept of conditional probability allows us to compute the probability of correctness of two integrated circuits given that the first integrated circuit in the pair was found to be faultless. It is important to note that this probability would be different than the probability of both integrated circuits being accepted if no information about the first integrated circuit acceptance is given. Conditional probabilities are widely used to characterize systems in probabilistic analysis and can be computed using the mathematical relation given in Eq. (1). In this section, we first provide a higher-order-logic formalization of the mathematical relation of Eq. (1) and then utilize this formal definition of conditional probability to verify some of its classical properties in the HOL theorem prover.

The definition of conditional probability, given in Eq. (1), can be formally expressed in higher-order logic as follows

Definition 4.1. Conditional probability

$\vdash_{def}$ ∀A B.
```
cond_prob A B =
if prob B = 0 then
   0
 else
   prob (A inter B) / prob B
```

by inheriting the higher-order-logic formalization of the probability function from [29]. The function `cond_prob` accepts two sets of infinite Boolean sequences A and B, corresponding to two events, and returns a real number that corresponds to the conditional probability of the first event A given the second event B has occurred. Eq. (1) is only valid for the case when $0 < Pr(B)$. This information is integrated in Definition 4.1 using the `if` statement, which allows us to assign a value of 0 to the conditional probability for the particular case when $Pr(B) = 0$. This way, we not only avoid the division by 0 scenario but also return a reasonable conditional probability result for the case when $Pr(B) = 0$. For example, if we consider the above mentioned example of testing a pair of integrated circuits, the probability of the event when we have both acceptable circuits given that the first tested circuit has been accepted with ($Pr(\textit{First circuit is acceptable}) = 0$) would be 0 according to our definition, which is actually the case.

Using the above definition, we now verify some classical properties of conditional probabilities [32] within the HOL theorem prover. The formal proofs for these properties not only ensure the correctness of our conditional probability definition but also play a vital role in reasoning about conditional probabilities of systems as will be seen in Section 5.

4.1. Conditional probability bounds

$$0 \leqslant Pr(A|B) \leqslant 1 \tag{2}$$

According to this property, the value of conditional probability always remains within the closed interval $[0, 1]$. It can be expressed in higher-order logic, using our conditional probability definition, as follows.

Theorem 4.1.

$\vdash$ ∀A B. (0 ⩽ cond_prob A B) ∧ (cond_prob A B ⩽ 1)

We proceed with the proof of this theorem by rewriting it with Definition 4.1 along with some simple arithmetic reasoning and splitting it into the following two subgoals.

¬(prob B = 0) ⟹ 0 ⩽ prob (A inter B)/prob B

¬(prob B = 0) ⟹ prob (A inter B)/prob B ⩽ 1

The first subgoal can now be proved using the probability axiom *Probability Bounds*, given in Table 2, along with some arithmetic reasoning. While the first step in the proof of the subgoal 1.2 is to move its denominator to the other side of the inequality as follows, using the given assumption and the *Probability Bounds* axiom.

¬(prob B = 0) ⟹ prob (A inter B) ⩽ prob B

The above subgoal can now be discharged using the *Probability Increasing* axiom, given in Table 2, since the set `A inter B` can be verified to be a subset of the set `B`. This also concludes the proof of Theorem 4.1 in HOL.

4.2. Conditional probability of the universal set

$$Pr(\mathbb{U}|B) = 1 \tag{3}$$

According to this property, the probability of the universal set (or the whole sample space) given that an event B has occurred is 1. It can be expressed in higher-order logic, using our conditional probability definition, as follows

Theorem 4.2.

$$\vdash \forall \mathtt{B}.\ (0 < \mathtt{prob\ B}) \implies (\mathtt{cond\_prob\ UNIV\ B} = 1)$$

This theorem can be verified in HOL by rewriting with Definition 4.1 and using some arithmetic and set theoretic reasoning. It is important to note that the property has been verified for the case when the probability of the given event B is greater than 0 as the theorem does not hold for the case when $Pr(B) = 0$.

4.3. Conditional probability is countably additive

If A_i is a sequence of mutually exclusive events then

$$Pr\left(\left(\bigcup_{i=1}^{\infty} A_i\right)\middle| B\right) = \sum_{i=1}^{\infty} Pr(A_i|B) \tag{4}$$

According to this property, conditional probability exhibits the countably additive property axiom just like the probability function. Theorems 1 and 2 along with this property show that conditional probability function exhibits all of the basic properties of ordinary probabilities and hence is a probability measure [32]. The countably additive property of conditional probability can be expressed in higher-order logic as follows.

Theorem 4.3.

$$\vdash \forall \mathtt{A\ B}.\ (\forall \mathtt{i\ j}.\ \neg(\mathtt{i} = \mathtt{j}) \implies \mathtt{disjoint\ (A\ i)(A\ j)}) \implies$$
$$(\mathtt{cond\_prob\ (bigunion(image\ A\ UNIV))\ B} = \mathtt{suminf\ (\lambda i.\ cond\_prob\ (A\ i)\ B)})$$

It is important to note here that variable A in the above theorem represents a sequence of sets rather than just a set like in the previous two theorems. Thus, the assumption in Theorem 4.3 ensures that the sets in this sequence A are mutually exclusive. The first argument of the function `cond_prob` corresponds to the set $(\bigcup_{i=1}^{\infty} A_i)$ as (`image A UNIV`) denotes the set $\{A_i \mid i \in N\}$, since `UNIV` is the countably infinite set of all possible *natural* numbers in this case. The function `bigunion`, explained in Table 1, then takes the union of all these sets in the set (`image A UNIV`). The RHS of the equality in Theorem 4.3 represents the infinite summation of the real sequence $(\lambda i.Pr(A_i|B))$ using the HOL function `suminf` [21], also explained in Table 1.

We proceed with the proof of this theorem by rewriting with the definition of the function `cond_prob`. This leads us to the following subgoal after some basic arithmetic simplification.

$$(\forall \mathtt{i\ j}.\ (\mathtt{i} = \mathtt{j}) \implies \mathtt{disjoint\ (A\ i)\ (A\ j)}) \implies$$
$$\mathtt{prob\ ((bigunion(image\ A\ UNIV))\ inter\ B)} = \mathtt{suminf\ (\lambda i.\ prob\ ((A\ i)\ inter\ B))}$$

Now, based on some already verified set theoretic principles in HOL, the set `((bigunion (image A UNIV)) inter B)` can be verified to be equal to `(bigunion (image (λi. A i inter B)) UNIV)`. This leads us to rewrite the above subgoal as follows.

$$(\forall \mathtt{i\ j}.\ \neg(\mathtt{i} = \mathtt{j}) \implies \mathtt{disjoint\ (A\ i)\ (A\ j)}) \implies$$
$$\mathtt{prob\ (bigunion\ (image\ (\lambda i.\ A\ i\ inter\ B)\ UNIV))} = \mathtt{suminf\ (\lambda i.\ prob\ ((A\ i)\ inter\ B))}$$

The `suminf` in the above subgoal can be verified to exist and to be equal to the expression on the left-hand side using the *Countable additivity* axiom of probability, formally proved by Hurd [29], since all sets (or events) in the set (`image (λi. A i inter B) UNIV`) are disjoint using the given assumption. This also concludes the HOL proof for Theorem 4.3.

4.4. Complement axiom of conditional probability

$$Pr(\bar{A}|B) = 1 - Pr(A|B) \tag{5}$$

According to this property, a conditional probability follows the complement axiom just like the probability function. It can be expressed in higher-order logic, using our conditional probability definition, as follows

Theorem 4.4.

```
⊢ ∀A B. 0 < prob B ⟹
  (cond_prob (compl A) B = 1 − cond_prob A B)
```

Rewriting the above theorem with Definition 4.1 along with some arithmetic reasoning we reach the following subgoal.

```
prob ((compl A)inter B) + prob (A inter B) = prob B
```

Since the two events on the left-hand-side (LHS) of the above subgoal are disjoint, the *probability additive* axiom, given in Table 2, can be used to rewrite the above subgoal as follows.

```
prob (((compl A)inter B) union (A inter B)) = prob B
```

Now based on set theoretic reasoning the set on the LHS of the above subgoal can be verified to be equivalent to the set B, which discharges the above subgoal from the goal stack and thus concludes the HOL proof for Theorem 4.4.

4.5. Difference axiom of conditional probability

$$Pr(A_1 - A_2|B) = Pr(A_1|B) - Pr(A_1 \cap A_2|B) \tag{6}$$

This property allows us to express the probability of an event $A_1 - A_2$ (difference between the sets A_1 and A_2) given some other event B without using the notion of set difference. It can be expressed in higher-order logic, using our conditional probability definition, as follows.

Theorem 4.5.

```
⊢ ∀A1 A2 B. cond_prob (A1 diff A2)B =
  cond_prob A1 B − cond_prob (A1 inter A2)B
```

Rewriting the above theorem with Definition 4.1 along with some arithmetic reasoning we reach the following subgoal.

```
prob ((A1 diff A2) inter B) + prob (A1 inter A2 inter B) =
  prob (A1 inter B)
```

Just like the proof of Theorem 4.3, the two events on the LHS of the above subgoal are disjoint and thus it can be verified using the *probability additive* axiom along with some set theoretic reasoning.

4.6. Union axiom of conditional probability

$$Pr(A_1 \cup A_2|B) = Pr(A_1|B) + Pr(A_2|B) - Pr(A_1 \cap A_2|B) \tag{7}$$

This property allows us to express the probability of an event $A_1 \cup A_2$ (union of the sets A_1 and A_2) given some other event B without using the notion of union. It can be expressed in higher-order logic, using our conditional probability definition, as follows.

Theorem 4.6.

```
⊢ ∀A1 A2 B. cond_prob (A1 union A2) B =
  cond_prob A1 B + cond_prob A2 B − cond_prob (A1 inter A2) B
```

We proceed with the proof of this theorem by first using Theorem 4.4 to rewrite the right-hand side (RHS) of the above goal as follows.

```
cond_prob (A1 union A2) B = cond_prob A2 B + cond_prob (A1 diff A2)B
```

Now, rewriting with Definition 4.1 along with some arithmetic reasoning we reach the following subgoal.

```
prob ((A1 union A2) inter B) =
  prob (A2 inter B) + prob ((A1 diff A2) inter B)
```

Like the previous two theorems, the two events on the RHS of the above subgoal are disjoint and thus it can now be verified using the *probability additive* axiom along with some set theoretic reasoning.

4.7. Multiplication rule of probability

$$Pr(A \cap B) = Pr(B)Pr(A|B) \tag{8}$$

Sometimes also referred to as the *theorem of compound probabilities*, the multiplication rule of probabilities is quite often found to be very useful for determining the probability that two events A and B will occur simultaneously. It can be expressed in higher-order logic, using our conditional probability definition, as follows.

Theorem 4.7.

```
⊢ ∀A B. prob (A inter B) = (prob B) * (cond_prob A B)
```

The first step for proving Theorem 4.7 in HOL is to rewrite with the definition of conditional probability. This step generates the following two subgoals after some arithmetic simplification.

```
¬(prob B = 0)  ⟹  prob (A inter B) =
  prob B * (prob (A inter B) / prob B)
(prob B = 0)  ⟹  prob (A inter B) = 0
```

The first subgoal can be proved in a very straightforward way using arithmetic reasoning. Whereas, the second subgoal can be proved based on the fact that $(A \cap B) \subset B$. Using this result along with the given assumption and the probability axioms *probability increasing* and *probability bounds*, given in Table 2, allows us to discharge the second subgoal from the HOL goal stack and thus conclude the proof of Theorem 4.7.

4.8. Total probability theorem

For a finite, mutually exclusive, and exhaustive sequence B_i of events and an event A,

$$Pr(A) = \sum_{i=1}^{m} Pr(B_i)Pr(A|B_i) \tag{9}$$

The *law of total probability* is a useful tool for breaking down the computation of a probability into a number of distinct cases. It can be expressed in higher-order logic, using our conditional probability definition, as follows.

Theorem 4.8.

```
⊢ ∀A B m. (bigunion (image B {n | n < m}) = UNIV) ∧
  (∀i j. i < m ∧ j < m ∧ ¬(i = j)  ⟹  disjoint (B i)(B j))  ⟹
    (prob A = sum (0, m) (λi. prob (B i) * cond_prob A (B i)))
```

The variable A in the above theorem represents an event, whereas B represents a sequence of sets. The first assumption ensures that the first m elements of the sequence B are exhaustive, i.e., their union gives the sample space or the universal set. Whereas, according to the second assumption in Theorem 4.8, the first m sets in sequence B are mutually exclusive. The RHS of the equality in Theorem 4.8 represents the summation of the first m terms of the real sequence $(\lambda i.Pr(B_i)Pr(A|B_i))$ using the HOL function `sum` [21], given in Table 1.

We verified Theorem 4.8 in HOL using case analysis on the variable m, i.e., splitting the goal into two subgoals for the cases when m is equal to 0 and $m + 1$.

```
(bigunion (image B {n | n < 0}) = UNIV)  ⟹  (prob A = 0)
(bigunion (image B {n | n < (m + 1)}) = UNIV) ∧
  (∀i j. i < (m + 1) ∧ j < (m + 1) ∧ ¬(i = j)  ⟹
    disjoint (B i) (B j))  ⟹
  (prob A = sum (0, (m + 1))′ (λi. prob (B i) ∗ cond_prob A(B i)))
```

The first subgoal can be verified because of the False assumption. While the step case can be verified using the definition of conditional probability, Theorem 4.7 and the *probability additive* axiom along with some set theoretic principles.

4.9. Bayes' law

$$Pr(A|B) = \frac{Pr(B|A)Pr(A)}{Pr(B)} \tag{10}$$

Bayes' law [32], named after the famous English philosopher Thomas Bayes, relates the conditional and marginal probabilities of two random events. In probabilistic analysis, posterior probabilities under some given observations are usually computed using Bayes' law. It can be expressed in higher-order logic, using our conditional probability definition, as follows.

Theorem 4.9.

```
⊢ ∀A B. (0 < prob B)  ⟹
  cond_prob A B = ((cond_prob B A) ∗ (prob A) / (prob B))
```

The first step for proving Theorem 4.9 in HOL is to rewrite the numerator using *multiplication rule of probability*, given in Theorem 4.7.

```
(0 < prob B)  ⟹  (cond_prob A B = (prob (B inter A)) / prob B)
```

The above subgoal can now be proved based on the definition of conditional probability and the commutativity property of intersection.

An alternate and more general form of Bayes' law can be expressed as follows [32]

$$Pr(A_i|B) = \frac{Pr(B|A_i)Pr(A_i)}{\sum_{i=1}^{m} Pr(B|A_i)Pr(A_i)} \tag{11}$$

for a finite sequence of exhaustive and mutually exclusive events A. This general form can also be verified in HOL using the proof steps of Theorem 4.9 along with the *total probability theorem*, given in Theorem 4.8. The corresponding HOL theorem is as follows.

Theorem 4.10.

```
⊢ ∀A B m k. (bigunion (image A {n | n < m}) = UNIV) ∧
  (∀i j. i < m ∧ j < m ∧ ¬(i = j)  ⟹  disjoint (A i)(A j)) ∧
  (0 < prob B) ∧ (k < m)  ⟹
    (cond_prob (A k) B =
      ((prob (A k) ∗ (cond_prob B (A k)))
      (sum (0, m) (λi. prob (A k) ∗ cond_prob B (A k)))))
```

This concludes our formalization of conditional probabilities. The verification of the above classical properties clearly indicates the correctness of our formal definition of conditional probability, given in Definition 4.1. Similarly, the formally verified theorems corresponding to the classical properties explicitly state all the conditions and requirements (assumptions) under which they hold, which is usually not the case when we look at the corresponding properties in paper-and-pencil verification based mathematical texts. These kind of assumptions play a vital part in the precise analysis of systems as they ensure the system correctness under the right set of constraints. The main contribution of the presented formalization is to pave the path for reasoning about conditional probabilities in a sound environment of a theorem prover. Our results can be directly built upon to formally reason about many safety-critical systems involving the domains of telecommunications,

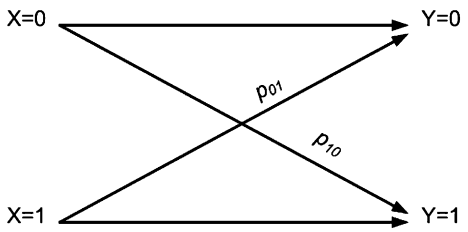

Fig. 1. Binary asymmetric channel with crossover probabilities.

computation algorithms and faults in hardware designs. For illustrating the utilization and effectiveness of the formalization, presented in this section, we utilize it to analyze a binary symmetric communication channel in the next section.

Our formalization can also be utilized to formalize more advance probabilistic analysis concepts like Markov chains. In this case, Definition 4.1 can be used to formalize the definition of Markov chain and then the formally verified properties of conditional probabilities, given in Theorems 4.1 to 4.10, can be used to verify Markov chains properties, such as, Joint probability, Chapman–Kolmogrove Equation and Absolute probability [7]. We are currently working on this extension and, once done, it will significantly broaden the horizon of theorem proving based formal probabilistic analysis by many folds.

5. Binary asymmetric channel analysis

A binary asymmetric channel is a noisy channel model for the case when the transmitter and receiver stations are capable of transmitting and receiving only two types of signals denoted by 0 and 1; usually termed as *bits*. Due to the channel noise there is a chance that the transmitted bit may be flipped when it reaches the receiver, i.e., a transmitted 0 is sometimes received as a 1 and a transmitted 1 is received as a 0. The error probabilities, usually referred to as the *crossover probabilities*, that govern the flipping of a bit are often known for a given channel in advance. Fig. 1 illustrates the binary asymmetric channel with p_{10} denoting the error probability when a transmitted 0 is flipped to a 1, $Pr(Y = 1|X = 0)$, during a transmission and p_{01} the error probability when a transmitted 1 is flipped to a 0, $Pr(Y = 0|X = 1)$, during transmission. If crossover probabilities are equal ($p_{10} = p_{01}$), then the channel is termed as the *binary symmetric channel*, which is a special case of the binary asymmetric channel.

Like the crossover probabilities, the probability of transmitting a 1 or a 0 is also known in advance as we usually have an idea of the transmitting patterns. On the other hand, probabilities such as the ones associated with the identity of the received bits, successful transmission and unsuccessful transmission are not known. These unknown probabilistic quantities play a vital role in estimating performance or reliability of the given channel and are thus obtained by conducting the probabilistic analysis of the binary asymmetric channel model, given in Fig. 1.

A binary asymmetric channel holds a significant place in probabilistic analysis of systems as being able to transmit effectively over the binary asymmetric channel can give rise to solutions for more complicated system behaviors. The examples of its usage in modeling and analyzing systems range from simple systems, such as a CD player reading from a scratched music CD, or a wireless cell phone capturing a weak signal from a far away relay tower, to more complicated ones, such as for complicated communication channels [48] or Bluetooth wireless networks [16]. Because of its widespread usage, a number of paper-and-pencil or simulation based probabilistic analysis of the binary asymmetric channel can be found in the literature. A couple of examples can be found in [46,2]. On the other hand, to the best of our knowledge, higher-order-logic theorem proving has never been used for the analysis of a binary channel mainly because of the inability to reason about conditional probabilities in a theorem prover so far. In this paper, we build upon the conditional probability theorems, presented in the previous section, to analyze the binary asymmetric channel model, given in Fig. 1. This exercise, not only illustrates the utilization of the theorems described in the previous section for conducting probabilistic analysis of systems but also, provides the foundational framework for the analysis of more complex systems that utilize the binary asymmetric channel model for their analysis.

The first step in the theorem proving based probabilistic analysis of the binary asymmetric channel is to formalize the problem in higher-order logic. We have four elements of randomness in this system: the transmitter output (X), receiver input (Y), error in bit 1 and error in bit 0. All these elements have two possible outcomes. Both X and Y can be 0 or 1. Similarly, the error during transmission could happen or not. These random elements can thus be modeled using the Bernoulli(p) random variable, which represents the coin flip experiment (has two possible outcomes) with the probability of *head* being p. Therefore, for the higher-order-logic modeling of the binary asymmetric channel, we utilize the higher-order-logic function for the Bernoulli random variable, `prob_bern`, given in [29] and modeled using the methodology summarized in Section 3. This function outputs a *True* with the probability equal to its first argument. The function `prob_bern` has been verified to be correct by proving its *probability mass function* (PMF) in the HOL theorem prover [29]. The corresponding higher-order-logic theorem is as follows.

Theorem 5.1.

$$\vdash \forall \mathtt{p}.\ 0 \leqslant \mathtt{p} \wedge \mathtt{p} \leqslant 1 \implies (\mathtt{prob}\,\{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{prob\_bern}\,\mathtt{p}\,\mathtt{s})\} = \mathtt{p})$$

It is important to note that the function prob_bern models the Bernoulli(p) and is more general than the Bernoulli($\frac{1}{2}$) random variable, described in Section 3, for which the success probability is always equal to 1/2. The transmitter output random variable (X) and the receiver input random variable (Y) are Bernoulli random variables that are equal to 1 with a known probability, say p_x and p_y, respectively. This behavior can be expressed in higher-order logic, using the formalized Bernoulli(p) random variable function, as follows.

Definition 5.1. Binary asymmetric channel transmitter/receiver

$$\vdash_{def} \forall \mathtt{p\_x}.\ \mathtt{bac\_tr\ p} = \mathtt{bind\ (prob\_bern\ p)\ (\lambda a.\ unit\ (if\ a\ then\ 1\ else\ 0))}$$

The function bac_tr accepts a *real* number p, which represents the probability of success for the Bernoulli(p) random variable, and returns a *natural* number, which is either equal to 0 or 1.

The output of the transmitter channel (X) can now be modeled by instantiating function bac_tr with probability p_x, whereas the input random variable to the receiver (Y) can be modeled by instantiating function bac_tr with probability p_y. It is important to note that, in the case of the receiver random variable, the exact probability for obtaining a 1 is not known upfront and needs to be computed in the analysis.

In the next few subsections, we utilize the above definition to formally specify various useful probabilistic quantities for the binary asymmetric channel in higher-order logic and verify their corresponding mathematical relations, using the theorems given in Section 4.

5.1. Probability of successful reception of a bit

We first verify the probability expressions for successful reception of a bit, i.e., the probability that bit x is received given that bit x was transmitted across the channel. These probabilities can be mathematically expressed as follows.

$$Pr(Y = 1|X = 1) = 1 - p_{01} \tag{12}$$

$$Pr(Y = 0|X = 0) = 1 - p_{10} \tag{13}$$

where p_{01} and p_{10} represent the conditional probabilities of receiving a 0 given that a 1 was transmitted and vice versa, respectively. The probabilistic relation for successfully receiving bit 1 can be expressed as a higher-order-logic theorem as follows.

Theorem 5.2.

$$\begin{aligned}
\vdash\ & \forall \mathtt{p_x\ p_y\ p_01}.\ 0 < \mathtt{p_x} \wedge \mathtt{p_x} < 1 \wedge \\
& (\mathtt{cond_prob}\ \{\mathtt{s} \mid \mathtt{fst\ (bac_tr\ p_y\ s)} = 0\} \\
& \qquad \{\mathtt{s} \mid \mathtt{fst\ (bac_tr\ p_x\ s)} = 1\} = \mathtt{p_01}) \implies \\
& (\mathtt{cond_prob}\ \{\mathtt{s} \mid \mathtt{fst\ (bac_tr\ p_y\ s)} = 1\} \\
& \qquad \{\mathtt{s} \mid \mathtt{fst\ (bac_tr\ p_x\ s)} = 1\} = 1 - \mathtt{p_01})
\end{aligned}$$

The first two assumptions in the above theorem ensure that the probability of transmitting a 1 from the transmitter, p_x, is greater than 0 and less than 1 and thus, omitting the chances of transmitting a sequence of 0's or 1's at all times. The third assumption represents the mathematical expression $Pr(Y = 0|X = 1) = p_{01}$, obtained from our reference model given in Fig. 1. The conclusion of the above theorem represents the mathematical expression $Pr(Y = 1|X = 1)$, which is the probability of successfully receiving the bit 1 given that bit 1 was transmitted.

Based on the given assumptions and the PMF of the Bernoulli random variable, given in Theorem 5.1, it can be verified that `0 < prob {s | fst (bac_tr p_x s) = 1}`. This result along with the *complement law of conditional probability*, verified in Theorem 4.4, can be used to rewrite Theorem 5.2 as follows.

$$\begin{aligned}
& 0 < \mathtt{p_x} \wedge \mathtt{p_x} < 1 \implies \\
& (\mathtt{cond_prob}\ \{\mathtt{s} \mid \mathtt{fst\ (bac_tr\ p_y\ s)} = 1\} \\
& \qquad \{\mathtt{s} \mid \mathtt{fst\ (bac_tr\ p_x\ s)} = 1\} = \\
& (\mathtt{cond_prob}\ (\mathtt{compl}\ \{\mathtt{s} \mid \mathtt{fst\ (bac_tr\ p_y\ s)} = 0\}) \\
& \qquad \{\mathtt{s} \mid \mathtt{fst\ (bac_tr\ p_x\ s)} = 1\}))
\end{aligned}$$

The new subgoal can now be discharged from the HOL goal stack by proving that the set `{s | fst (bac_tr p_y s) = 1}` is equivalent to the set `(compl {s | fst (bac_tr p_y s) = 0})` using the definition of the function `bac_tr` along with some set theoretic principles. This also concludes the proof for Theorem 5.2. Similarly, the successful reception probability for the bit 0, given in Eq. (13), can also be formally verified.

5.2. Received bit probabilities

In this section, we formally verify the following probability expressions for receiving a 1 or a 0 at the receiver of the binary asymmetric channel.

$$Pr(Y = 1) = p_{10}(1 - p_x) + (1 - p_{01})p_x \tag{14}$$

$$Pr(Y = 0) = (1 - p_{10})(1 - p_x) + p_{01}p_x \tag{15}$$

The probability of receiving a 1, which is actually equal to the unknown probability p_y based on the result of Theorem 5.1 and is given in Eq. (14) above, can be expressed in higher-order logic as follows.

Theorem 5.3.

```
⊢ ∀p_x p_y p_01 p_10. 0 < p_x ∧ p_x < 1 ∧
    (cond_prob {s | fst (bac_tr p_y s) = 0}
               {s | fst (bac_tr p_x s) = 1} = p_01) ∧
    (cond_prob {s | fst (bac_tr p_y s) = 1}
               {s | fst (bac_tr p_x s) = 0} = p_10)  ⟹
    (prob {s | fst (bac_tr p_y s) = 1} =
               p_10 * (1 − p_x) + (1 − p_01) * p_x)
```

We proceed with the proof of this theorem by rewriting and simplifying the proof goal using the successful transmission probability of bit 1, given in Theorem 5.2, and the PMF of the Binomial random variable, given in Theorem 5.1, as follows.

```
prob {s | fst (bac_tr p_y s) = 1} =
    (cond_prob {s | fst (bac_tr p_y s) = 1}
               {s | ¬fst (bac_tr p_x s) = 1}) *
         (prob{s | ¬fst (bac_tr p_x) = 1}) +
    (cond_prob {s | fst (bac_tr p_y s) = 1}
               {s | fst (bac_tr p_x s) = 1}) *
         (prob {s | fst (bac_tr p_x) = 1})
```

The statement of the above subgoal is very closely related to the *total probability theorem*, given in Theorem 4.8. Therefore, Theorem 4.8 can be used to verify the above subgoal if the expression on its RHS can be represented as a summation of real sequence. This desired form for the RHS can be obtained using some simple arithmetic reasoning and is given below.

```
prob {s | fst (bac_tr p_y s) = 1} =
    sum (0, 2)
    (λi.
       prob
            ((λm.
            (if m = 0 then
               {s | ¬fst (bac_tr p_x s) = 1}
            else
               (if m = 1 then
```

```
                {s | fst (bac_tr p_x s) = 1}
            else
                {s | F})) i) *
    cond_prob {s | fst (bac_tr p_y s) = 1}
        ((λm.
        (if m = 0 then
            {s | ¬fst (bac_tr p_x s) = 1}
        else
            (if m = 1 then
                {s | fst (bac_tr p_x s) = 1}
            else
                {s | F})) i))
```

Now, Theorem 4.8 can be used to discharge the above goal from the HOL goal stack by verifying the mutually exclusiveness and exhaustiveness of the given real sequence based on some arithmetic and set theoretic reasoning. This step concludes the verification of Theorem 5.3. Following similar reasoning as above, we also verified the probability expression of receiving a 0 at the receiver of the binary asymmetric channel, given in Eq. (15).

5.3. Probability of successful transmission of a bit

In this section, we verify the following probability expressions for successful transmission of a bit, i.e., the probability that bit x is transmitted given that bit x was received from the channel.

$$Pr(X = 1|Y = 1) = \frac{(1 - p_{10})p_x}{p_{10}(1 - p_x) + (1 - p_{01})p_x} \tag{16}$$

$$Pr(X = 0|Y = 0) = \frac{(1 - p_{10})(1 - p_x)}{(1 - p_{10})(1 - p_x) + p_{01}p_x} \tag{17}$$

The probabilistic relation for successfully transmitting bit 1 across the binary asymmetric channel can be expressed as a higher-order-logic theorem as follows.

Theorem 5.4.

```
⊢ ∀p_x p_y p_01 p_10. 0 < p_x ∧ p_x < 1 ∧
    (cond_prob {s | fst (bac_tr p_y s) = 0}
               {s | fst (bac_tr p_x s) = 1}' = p_01) ∧
    (cond_prob {s | fst (bac_tr p_y s) = 1}
               {s | fst (bac_tr p_x s) = 0} = p_10) ∧
    (0 < p_10 ∨ p_01 < 1)  ⟹
        (cond_prob {s | fst (bac_tr p_x s) = 1}
                   {s | fst (bac_tr p_y s) = 1} =
    (1 − p_01) * p_x ∧ (p_10 * (1 − p_x) + (1 − p_01) * p_x))
```

A new assumption `(0 < p_10 ∨ p_01 < 1)` has been added in the above theorem besides the ones used previously. This assumption is used to prevent the scenario when bit 1 is never received. The conclusion of the above theorem represents the mathematical expression $Pr(X = 1|Y = 1)$, which is the desired probability of transmission of bit 1 given that bit 1 was received.

The denominator of the expression on the RHS of Theorem 5.4 is basically the probability of receiving a bit 1, as given by Theorem 5.3. Whereas, the first expression in the numerator of Theorem 5.4, `(1 - p_01)`, is the probability of successful reception of bit 1, given in Theorem 5.2. Thus, rewriting with Theorems 11, 12 and 14 we get the following simplified version of the given theorem.

$$
\begin{aligned}
&0 < \mathtt{p\_x} \wedge \mathtt{p\_x} < 1 \wedge (0 < \mathtt{p\_10} \vee \mathtt{p\_01} < 1) \implies \\
&\quad (\mathtt{cond\_prob}\,\{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_x}\,\mathtt{s}) = 1\} \\
&\qquad \{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_y}\,\mathtt{s}) = 1\} = \\
&\quad (\mathtt{cond\_prob}\,\{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_y}\,\mathtt{s}) = 1\} \\
&\qquad \{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_x}\,\mathtt{s}) = 1\}) * \\
&\qquad\quad (\mathtt{prob}\,\{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_x}\,\mathtt{s}) = 1\}) \\
&\qquad\qquad (\mathtt{prob}\,\{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_y}\,\mathtt{s}) = 1\}))
\end{aligned}
$$

The above subgoal can now be verified by instantiating Bayes' law, verified in Theorem 4.10, with the two sets involved. This also concludes the proof for Theorem 5.4. Similarly, we also verified the successful transmission probability expression for bit 0, given in Eq. (17).

5.4. Channel error probability

Finally, we formally verify an expression for the probability of error in the binary asymmetric channel model. Error occurs when a transmitted 1 is received as a 0 or a transmitted 0 is received as a 1 and thus the corresponding mathematical expression is as follows.

$$Pr\big((Y = 0 \wedge X = 1) \vee (Y = 1 \wedge X = 0)\big) = p_{01}\mathtt{p\_x} + p_{10}(1 - p_x) \tag{18}$$

This probability can be formalized using our definitions as follows.

Theorem 5.5.

$$
\begin{aligned}
&\vdash \forall \mathtt{p\_x}\,\mathtt{p\_y}\,\mathtt{p\_01}\,\mathtt{p\_10}.\ 0 < \mathtt{p\_x} \wedge \mathtt{p\_x} < 1 \wedge \\
&\quad (\mathtt{cond\_prob}\,\{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_y}\,\mathtt{s}) = 0\} \\
&\qquad\qquad \{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_x}\,\mathtt{s}) = 1\} = \mathtt{p\_01}) \wedge \\
&\quad (\mathtt{cond\_prob}\,\{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_y}\,\mathtt{s}) = 1\} \\
&\qquad\qquad \{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_x}\,\mathtt{s}) = 0\} = \mathtt{p\_10}) \implies \\
&\quad (\mathtt{prob}\,\{\mathtt{s} \mid (\mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_y}\,\mathtt{s})' = 0 \wedge \\
&\qquad\qquad\quad \mathtt{fst}(\mathtt{bac\_tr}\,\mathtt{p\_x}\,\mathtt{s}) = 1) \vee \\
&\qquad\qquad\quad (\mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_y}\,\mathtt{s}) = 1 \wedge \\
&\qquad\qquad\quad \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_x}\,\mathtt{s}) = 0)\} = \\
&\qquad\qquad \mathtt{p\_01} * \mathtt{p\_x} + \mathtt{p\_10} * (1 - \mathtt{p\_x}))
\end{aligned}
$$

We can use the *probability additive axiom*, due to the disjoint nature of the events in the set on the LHS of the conclusion, along with the *multiplication rule of probability*, verified in Theorem 4.7, and some set theoretic principles to simplify the proof goal of Theorem 5.5 as follows.

$$
\begin{aligned}
&0 < \mathtt{p\_x} \wedge \mathtt{p\_x} < 1 \wedge \\
&\quad (\mathtt{cond\_prob}\,\{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_y}\,\mathtt{s}) = 0\} \\
&\qquad\qquad \{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_x}\,\mathtt{s}) = 1\} = \mathtt{p\_01}) \wedge \\
&\quad (\mathtt{cond\_prob}\,\{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_y}\,\mathtt{s}) = 1\} \\
&\qquad\qquad \{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_x}\,\mathtt{s}) = 0\} = \mathtt{p\_10}) \implies \\
&\quad ((\mathtt{prob}\,\{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_x}\,\mathtt{s}) = 1\}) * \\
&\quad (\mathtt{cond\_prob}\,\{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_y}\,\mathtt{s}) = 0\} \\
&\qquad\quad \{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_x}\,\mathtt{s}) = 1\}) + \\
&\quad (\mathtt{prob}\,\{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_x}\,\mathtt{s}) = 0\}) * \\
&\quad (\mathtt{cond\_prob}\,\{\mathtt{s} \mid \mathtt{fst}\,(\mathtt{bac\_tr}\,\mathtt{p\_y}\,\mathtt{s}) = 1\}
\end{aligned}
$$

$$\{s \mid \texttt{fst}\,(\texttt{bac\_tr}\ \texttt{p\_x}\ s) = 0\}) =$$
$$\texttt{p\_01} * \texttt{p\_x} + \texttt{p\_10} * (1 - \texttt{p\_x}))$$

This subgoal can now be discharged from the HOL goal stack by simplifying based on the given assumptions and the PMF of the Bernoulli random variable, given in Theorem 5.1, along with some arithmetic reasoning. This also concludes the verification of Theorem 5.5.

The above results clearly demonstrate the effectiveness of the proposed theorem proving based probabilistic analysis approach for reasoning about conditional probabilities. Due to the formal nature of the model and inherent soundness of theorem proving, we have been able to verify the probabilistic properties of interest regarding the given binary asymmetric channel with 100% precision; a novelty which is not available in simulation. Similarly due to the high expressiveness of higher-order logic, we have been able to verify generic properties that are valid for all values of cross-over and transmission probabilities; something that cannot be done in probabilistic model checking. For instance, the proposed approach is also superior than the paper-and-pencil proof methods in a way as the chances of making human errors, missing critical assumptions and proving wrongful statements are almost nil since all proof steps are applied within the sound core of the HOL theorem prover. These additional benefits come at the cost of the time and effort spent, while constructing the formal model of the system and formally reasoning about its properties, by the user. But, the formally verified classical conditional probability related properties, presented in this paper, lead to a significant reduction in the interactive verification effort. For example, the binary asymmetric channel analysis, presented in this section, only consumed approximately 40 man-hours and 800 lines of HOL code by an expert user, mainly because the analysis utilizes the theorems given in Section 4.

6. Conclusions

Conditional probability is one of the most widely used probabilistic analysis concepts. It allows us to estimate probabilities when some partial information concerning the result of the experiment is available. This paper presents an infrastructure that can be used to reason about conditional probabilities in a higher-order-logic theorem prover, which to the best of our knowledge was not possible as of now. The primary motivation behind this work is to augment the higher-order-logic theorem proving based probabilistic analysis with this new capability in order to be able to precisely analyze a broader range of systems and properties. The precision and accuracy of the probabilistic analysis results attained with this approach in turn proves to be very useful for the performance and reliability optimization of safety critical and highly sensitive engineering and scientific applications.

In this paper, we provided a higher-order logic definition of the mathematical concept of conditional probability. Then, we utilized this definition to verify some classical conditional probability properties in a higher-order logic theorem prover (HOL). It is important to note here that the ideas presented in this paper are not specific to the HOL theorem prover and can be adapted to any other higher-order-logic theorem prover as well, such as Isabelle [40], Coq [13] or PVS [42]. The formally verified conditional probability properties not only ensure the correctness of the proposed conditional probability definition but also play a vital part in reasoning about conditional probabilities while conducting probabilistic analysis of systems. In order to illustrate the practical effectiveness of these formally verified properties, we illustrated their usage for the probabilistic analysis of a binary asymmetric channel.

The formalization and verification results, presented in this paper, are quite general and thus can serve as a foundational building block for the development of many theorem proving based probabilistic analysis techniques and can be utilized for the precise analysis of a wide range of engineering and scientific applications. Markov chains [49] is one of the most commonly used probabilistic modeling method in areas like Physics, Queueing Theory, Internet applications and statistical testing. The basic definition of a Markov chain is based on the concept of conditional probabilities and thus our results can be built upon to develop an infrastructure for representing and analyzing Markov chains in a theorem prover. Another useful probabilistic approach that relies heavily on the concept of conditional probabilities, and thus can be developed in a theorem prover based on the work presented in this paper, is computer system reliability analysis [3]. Similarly, our results can be utilized for the formalization of some fundamental probabilistic analysis concepts, such as conditional probability distribution, conditional independence, conditional expectation and conditional variance [43]. The formalization infrastructure, presented in this paper, can also be used as is for the analysis of a number of interesting applications, such as communication channels [48,16], algorithms for computation problems like the Best Prize problem [43], and fault detection schemes for hardware designs [14,31].

For our verification, we utilized the HOL theories of *Boolean algebra*, *sets*, *lists*, *natural* and *real* numbers, *real analysis*, *measure* and *probability*. This set of theories is a good representation of the state-of-the-art in formalized mathematics. Thus, our results can be considered as a useful case study for demonstrating the capabilities of automated reasoning and illustrating its usefulness. Summarizing our experiences as far as this exercise is concerned, we can say that formalizing mathematics in a mechanical system is a tedious work that requires deep understanding of both mathematical concepts and theorem-proving. The HOL automated reasoners aid somewhat in the proof process by automatically verifying some of the first-order-logic goals but most of the times we had to guide the tool by providing the appropriate rewriting and simplification rules. On the other hand, we found theorem-proving very helpful in book keeping. Another major advantage of theorem proving is that once the proof of a theorem is established, due to the inherent soundness of the approach, it is guaranteed to be valid and the complete proof details can be readily accessed, contrary to the case of paper-pencil proofs

where we may have to explore an enormous amount of mathematical literature to find proof details. For example, in the proofs presented in this paper, we utilized theorems from mathematical theories but still the detailed proof steps can be readily traced down to the level of 5 basic axioms or 8 primitive inference rule of the HOL system, whereas, it is very hard to find one text book that provides the detailed mathematical proofs of all the mathematical facts that are utilized in our proofs. Thus, it can be concluded that theorem-proving is a tedious but promising field, which can help mathematicians to cope with the explosion in mathematical knowledge and to save mathematical concepts from corruption. Also, there are areas, such as security critical software, in military or medicine applications for example, where theorem-proving will soon become a dire need.

References

[1] L. de Alfaro, Formal verification of probabilistic systems, PhD thesis, Stanford University, Stanford, USA, 1997.
[2] P. Amblard, O. Michel, S. Morfu, Revisiting the asymmetric binary channel: joint noise enhanced detection and information transmission through threshold devices, in: Noise in Complex Systems and Stochastic Dynamics III, in: SPIE, vol. 5845, 2005, pp. 50–60.
[3] R. Apthorpe, A probabilistic approach to estimating computer system reliability, in: Proceedings of the 15th USENIX Conference on System Administration, USENIX Association, 2001, pp. 36–46.
[4] P. Audebaud, C. Paulin-Mohring, Proofs of randomized algorithms in Coq, in: Mathematics of Program Construction, in: LNCS, vol. 4014, Springer, 2006, pp. 49–68.
[5] C. Baier, B. Haverkort, H. Hermanns, J. Katoen, Model checking algorithms for continuous time Markov chains, IEEE Transactions on Software Engineering 29 (4) (2003) 524–541.
[6] N. Benton, J. Hughes, E. Moggi, Monads and effects, in: International Summer School on Applied Semantics, in: LNCS, vol. 2395, Springer, 2000, pp. 42–122.
[7] R. Bhattacharya, E. Waymire, Stochastic Processes with Applications, John Wiley and Sons, 1990.
[8] J. Bialas, The σ-additive measure theory, Journal of Formalized Mathematics 2 (1990).
[9] Boeing 777 Incident, http://aviation-safety.net/database, 2009.
[10] C. Chang, R. Lee, Symbolic Logic and Mechanical Theorem Proving, Academic Press, 1973.
[11] A. Church, A formulation of the simple theory of types, Journal of Symbolic Logic 5 (1940) 56–68.
[12] E. Clarke, O. Grumberg, D. Peled, Model Checking, The MIT Press, 2000.
[13] Coq, http://pauillac.inria.fr/coq/, 2009.
[14] W. Daehn, Load balancing in a hybrid ATPG environment, IEEE Transactions on Computers 40 (7) (1991) 878–882.
[15] L. Devroye, Non-Uniform Random Variate Generation, Springer, 1986.
[16] N. Golmie, R.V. Dyck, A. Soltanian, Interference of bluetooth and IEEE 802.11: Simulation modeling and performance evaluation, in: Proc. International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, ACM, 2001, pp. 11–18.
[17] M. Gordon, Mechanizing programming logics in higher-order logic, in: Current Trends in Hardware Verification and Automated Theorem Proving, Springer, 1989, pp. 387–439.
[18] M. Gordon, T. Melham, Introduction to HOL: A Theorem Proving Environment for Higher-Order Logic, Cambridge University Press, 1993.
[19] A. Gupta, Formal hardware verification methods: A survey, Formal Methods in System Design 1 (2–3) (1992) 151–238.
[20] J. Harrison, Formalized mathematics, Technical Report 36, Turku Centre for Computer Science, Finland, 1996.
[21] J. Harrison, Theorem Proving with the Real Numbers, Springer, 1998.
[22] O. Hasan, S. Tahar, Formalization of the continuous probability distributions, in: Automated Deduction, in: LNAI, vol. 4603, Springer, 2007, pp. 3–18.
[23] O. Hasan, S. Tahar, Verification of expectation properties for discrete random variables in HOL, in: Theorem Proving in Higher-Order Logics, in: LNCS, vol. 4732, Springer, 2007, pp. 119–134.
[24] O. Hasan, S. Tahar, Verification of probabilistic properties in HOL using the cumulative distribution function, in: Integrated Formal Methods, in: LNCS, vol. 4591, Springer, 2007, pp. 333–352.
[25] O. Hasan, S. Tahar, Performance analysis of ARQ protocols using a theorem prover, in: Proc. International Symposium on Performance Analysis of Systems and Software, IEEE Computer Society, 2008, pp. 85–94.
[26] O. Hasan, S. Tahar, Formal verification of tail distribution bounds in the HOL theorem prover, Mathematical Methods in the Applied Sciences 32 (4) (2009) 480–504.
[27] O. Hasan, S. Tahar, Performance analysis and functional verification of the stop-and-wait protocol in HOL, Journal of Automated Reasoning 42 (1) (2009) 1–33.
[28] O. Hasan, S. Tahar, Performance analysis of wireless systems using theorem proving, Electronic Notes in Theoretical Computer Science 242 (2) (2009) 43–58.
[29] J. Hurd, Formal verification of probabilistic algorithms, PhD thesis, University of Cambridge, UK, 2002.
[30] J. Hurd, Verification of the Miller Rabin probabilistic primality test, Logic and Algebraic Programming 56 (1) (2003) 3–21.
[31] D. Jin, H. Jiandongl, A new probabilistic approach for estimating fault detection probability, Journal of Electronics 11 (4) (2007) 309–314.
[32] R. Khazanie, Basic Probability Theory Applications, Goodyear, 1976.
[33] M. Kwiatkowska, G. Norman, D. Parker, Quantitative analysis with the probabilistic model checker PRISM, Electronic Notes in Theoretical Computer Science 153 (2) (2005) 5–31.
[34] D. MacKay, Introduction to Monte Carlo methods, in: Learning in Graphical Models, in: NATO Science Series, Kluwer Academic Press, 1998, pp. 175–204.
[35] Mars Climate Orbiter, http://solarsystem.nasa.gov/missions, 2008.
[36] Mars Polar Lander, http://mpfwww.jpl.nasa.gov/msp98/, 2009.
[37] R. Milner, A theory of type polymorphism in programming, Journal of Computer and System Sciences 17 (1977) 348–375.
[38] A. Nedzusiak, σ-Fields and probability, Journal of Formalized Mathematics 1 (1989).
[39] D. Parker, Implementation of symbolic model checking for probabilistic system, PhD thesis, University of Birmingham, UK, 2001.
[40] L. Paulson, Isabelle: A Generic Theorem Prover, LNCS, vol. 828, Springer, 1994.
[41] L. Paulson, ML for the Working Programmer, Cambridge University Press, 1996.
[42] PVS, http://pvs.csl.sri.com, 2009.
[43] S. Ross, Simulation, Academic Press, 2002.
[44] J. Rutten, M. Kwaiatkowska, G. Normal, D. Parker, Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems, CRM Monograph Series, vol. 23, American Mathematical Society, 2004.
[45] K. Sen, M. Viswanathan, G. Agha, VESTA: A statistical model-checker and analyzer for probabilistic systems, in: Proc. IEEE International Conference on the Quantitative Evaluation of Systems, 2005, pp. 251–252.

[46] R. Silverman, On binary channels and their cascades, IRE Transactions on Information Theory 1 (3) (1955) 19–27.
[47] K. Tridevi, Probability and Statistics with Reliability, Queuing and Computer Science Applications, Wiley–Interscience, 2002.
[48] H. Wang, N. Moayeri, Finite-state Markov channel—a useful model for radio communication channels, Transactions on Vehicular Technology 44 (1) (1995) 163–171.
[49] R. Yates, D. Goodman, Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers, Wiley, 2005.